How Do People Really Use Text Editors?

John Whiteside, Norman Archer, Dennis Wixon, and Michael Good

Digital Equipment Corporation
Maynard, Massachusetts 01754

Abstract

Keystroke statistics were collected on editing systems while people performed their normal work. Knowledge workers used an experimental editor, and secretaries used a word processor. Results show a consistent picture of free use patterns in both settings. Of the total number of keystrokes, text entry accounted for approximately 1/2, cursor movement for about 1/4, deletion for about 1/8, and all other functions for the remaining 1/8. Analysis of keystroke transitions and editing states is also presented. Implications for past research, editor design, keyboard layout, and benchmark tests are discussed.

1. Introduction

Different approaches have been used to study usage of text editors. Several researchers have proposed models of user performance and then tested user behavior with respect to these models [1, 2, 3, 4, 5]. A second approach has been to compare editors by having subjects perform specific benchmark tasks [9]. A third approach has been to compare command transition matrices for various editors [6].

1.1. Ideal Models

Models of text editor performance have focused either on predicting the time which users would take to perform a particular task, or on the strategy which users would employ to accomplish a particular goal. Models which focus on time [1, 2, 3] treat the overall task time as the sum of the times which various subtasks take. These subtasks include operations such as typing, pointing, thinking, and waiting for system response. Models which focus on commands include the GOMS model [1], Hammer's constrained optimal model [5], and Embley and Nagy's model based on file comparison [3].

All of these models share a common attribute of concentrating on idealized performance. For example, Card et al. ignore errors, give subjects practice problems, and instruct them with respect to how particular activities are to be performed. Alternatively, the performance of users is considered as a deviation from the model. Such deviation is interpreted as suggesting the need for user training [5] or has been tabulated and classified according to user experience [3]. Hammer found that an optimal model predicted user performance rather poorly. Constraining the model by taking into account only those commands which were actually used improved the fit considerably.

1.2. Benchmark Tasks

One of the most detailed analyses of editors was conducted by Roberts [9]. She developed methods for comparing editors on the basis of speed of use, ease of learning, range of functionality, and error-proneness. Speed analyses of editors involved the performance of benchmark tasks by experts on each system. She used these methods to compare the TECO, Wylbur, NLS, and Wang text editors. TECO and Wylbur were found to be significantly slower to use than the other two editors. TECO was also significantly harder to learn than the other three editors.

1.3. Transition Matrices

Hammer and Rouse [6] analyzed free keystroke records for both programmers and documenters using either SOS or TECO. Behavior within an editor was categorized into uniform editor primitives, such as type one line, alter one character, type many lines, and alter many characters. The matrices of transition probabilities between editor primitives were then compared to an overall transition matrix

---

VAX, VMS, VT100, DEC, and PDP are trademarks of Digital Equipment Corporation.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1982 ACM 0-89791-075-3/82/006/0029 $00.75

which represented the average of various editors. The results led Hammer and Rouse to conclude that the differences between editors could have been due to the fact that different people used them. In another analysis, Hammer and Rouse compared the same users editing programs and documents and found that about 75% of the subjects used similar techniques for each. All of these analyses were applied to editors which use hard copy terminals.

1.4. Present Approach

Instead of postulating an ideal editing model, we attempted to record and describe actual usage patterns. To this end keystroke-level data were collected while people performed their normal work on either an experimental editor or a word processing editor. Unlike previous research, this free-use logging was collected solely from screen editors and analyzed at several levels. Recommendations for editor design and evaluation procedures are based on actual editor usage.

2. Method

In the present experiment, users' keystrokes were logged and time-stamped for two editors. The first editor was an experimental editor called the Editor Prototyping Tool (EPT). The second editor was Digital Equipment Corporation's 200 Series commercial word processor (WPS). Both of these are character-oriented screen editors, but they differ in important ways.

2.1. EPT Description

EPT was designed specifically for research purposes. Thus, it contained a built-in keystroke-logging facility, designed not to influence software performance. EPT provided a simple set of editing functions including cursor movement anywhere on the screen, search, text relocation, and text deletion. Typing could be done in either insert or overstrike mode. These functions were activated by dedicated function keys; in most cases by a single keystroke. A short help facility was provided.

EPT contained few of the formatting functions present in a normal word processor; however it did perform automatic word wrapping.

2.2. WPS Description

The WPS word processor is a character-oriented, screen editor in which editing is done on the bottom line of the screen. The cursor may be moved within the boundaries of existing text. The keyboard has a variety of special keys for editing and cursor movement. A special "gold" key on the keypad, when used in combination with other keypad keys and with keys on the regular keyboard, greatly expands the number of possible functions which may be selected directly. WPS has logical equivalents for all EPT functions, and has many more functions in addition, including sophisticated text formatting functions.

2.3. Subjects

Six subjects contributed to the EPT work sample. These subjects were knowledge workers creating and revising their own reports and memos. The logging of data was not begun until all subjects had at least three weeks experience with EPT, which was sufficient time to allow these subjects to learn the features of this simple editor.

The subjects using WPS were eight secretaries who were carrying out regular secretarial and administrative duties for a research and development department of Digital Equipment Corporation. The experience of the subjects varied widely, with three having had almost two years of experience with the system and the full set of three training courses. One had a year experience and one training course. The remaining four subjects had no training course but were self taught, with experience of one to six months on the system.

2.4. Keystroke Logging

As mentioned above, EPT had a built-in logging facility. A permanent record of every keystroke was made, together with time of occurrence. EPT ran under the VMS timesharing system on a VAX-11/780 computer, supporting DEC VT100 terminals.

For the WPS word processing system, character streams and occurrence times from six VT100W terminals were collected by a PDP 11/34 computer, in such a way that WPS system performance was not affected. The recording was done over a period of 11 days.

3. Results

3.1. General Characteristics of the Work Samples

Table 1 shows some summary characteristics of the work samples.

	EPT	WPS
Total person-hours in sample	212	482
Hours taking breaks (over 150 sec)	127	390
Hours active work	86	92
Total keystrokes	510,503	406,102

Table 1: Comparison of EPT and WPS Work Samples

The samples contain roughly equal numbers of keystrokes, about half a million each. Total person hours in the WPS

Keys	Percent of keystrokes	Percent of time	Time per keystroke
Printing characters	53.58	46.84	0.53
Left-Arrow	12.90	5.08	0.24
Right-Arrow	12.52	5.49	0.27
Delete	7.35	7.38	0.61
Down-Arrow	5.67	9.89	1.06
Up-Arrow	3.15	5.89	1.14
Return	1.38	4.38	1.93
Tab	1.01	1.66	0.99
Word	0.34	0.39	0.71
Insert/Overstrike	0.33	1.02	1.86
Do It	0.28	0.97	2.14
Backup	0.24	0.54	1.36
Next Screen	0.22	2.34	6.38
Next Line	0.21	0.67	1.93
Advance	0.17	0.42	1.53
Exit	0.14	2.69	11.89
Last Screen	0.10	1.02	6.15
Erase	0.10	0.59	3.76
Search	0.07	0.66	6.17
Paragraph	0.05	0.17	2.23
Move	0.04	0.17	2.38
Cancel	0.03	0.28	5.92
Back Space	0.03	0.07	1.64
Copy	0.02	0.18	4.53
Goto Page	0.02	0.43	11.28
Sentence	0.02	0.43	12.80
Help	0.02	0.15	5.66
Command	0.01	0.13	7.74
Page	0.01	0.04	2.86
Totals	100.00	100.00	0.61

Table 2: EPT Keystroke Frequency Table

sample were over twice as many as in the EPT sample; this discrepancy was due to the secretaries' habit of turning on their machines in the morning and leaving them running for the entire day. EPT ran on a general timesharing system. so users were less likely to leave the editor running constantly. This difference can be seen by looking at the number of hours of inactivity (defined as the sum of all periods of greater than 150 seconds without a keystroke) which is much larger for the WPS sample. The remaining time, hours of active work, is comparable for both samples at somewhat under 100 hours.

3.2. Individual Keystroke-Level Analysis

Table 2 shows summary statistics for keystroke usage for the EPT editor. The left-hand column identifies the function. Each entry refers to an individual key, except for the case of "printing characters," which includes all alphanumerics and symbols under a single heading. The keys are ordered by frequency of usage, in terms of percentage of total keystrokes recorded. This number is shown in the second column; the percentages are based on the total number of keystrokes in the work sample (510,503). The remaining two columns in Table 2 show data with respect to time. The percentage of use by time is shown in the third column, and the mean time per keystroke is shown in the fourth column. In computing

Keys	Percent of keystrokes	Percent of time	Time per keystroke
Printing characters	54.81	47.48	0.71
Advance	15.48	11.91	0.63
Line	10.30	6.27	0.50
Backup	6.59	7.41	0.92
Rub Character	3.33	3.35	0.82
Return	2.83	8.08	2.33
Tab	1.52	1.79	0.96
Word	1.24	0.65	0.42
Delete Character	1.11	0.90	0.66
Paragraph	0.44	0.86	1.57
Rub Word	0.24	0.46	1.53
Menu	0.23	2.69	9.66
Page	0.21	0.97	3.71
Delete Word	0.19	0.19	0.83
Upper Case	0.17	0.11	0.51
File Document	0.12	1.61	10.50
Select	0.13	0.38	2.40
Ruler	0.12	0.60	4.18
Manual Pagination	0.08	0.78	8.24
Date & Time	0.07	0.20	2.46
New Page	0.06	0.34	4.30
Underline	0.06	0.11	1.40
Gold Key	0.05	0.02	0.25
< >	0.05	0.09	1.60
Halt	0.04	0.11	1.96
Cut	0.04	0.12	2.36
Tab Pos	0.04	0.13	2.55
Top of Document	0.04	0.53	11.45
Rub Line	0.03	0.03	0.80
Control characters	0.03	0.11	2.71
Lower Case	0.03	0.04	1.11
Swap	0.03	0.03	0.81
Sentence	0.02	0.06	1.89
Search	0.02	0.28	9.43
View	0.02	0.14	4.78
Gold Paste	0.02	0.05	1.69
Center	0.02	0.04	1.37
Paste	0.02	0.10	4.79
Get Document	0.01	0.17	9.56
Bottom of Document	0.01	0.40	22.95
Advance to End	0.01	0.12	7.74
Rub Sentence	0.01	0.02	1.11
Hyphen Push	0.01	0.01	1.08
Cont Search	0.01	0.02	2.13
Backup to End	0.01	0.13	17.78
Subscript	0.01	0.01	1.80
Paragraph Marker	0.00	0.01	1.62
Superscript	0.00	0.02	2.53
Command	0.00	0.00	3.75
Hyphen Pull	0.00	0.01	0.64
Bold	0.00	0.00	1.33
Remove Underline	0.00	0.01	0.48
Dead Key	0.00	0.02	3.34
User Defined Key	0.00	0.01	8.17
Page Marker	0.00	0.00	6.62
Cont Search & Sel	0.00	0.00	1.12
Gold Cut	0.00	0.00	1.16
Undelete Word	0.00	0.00	3.18
Replace	0.00	0.00	1.51
Library	0.00	0.00	2.34
Print Hyphen	0.00	0.00	0.00
Abbreviation	0.00	0.00	0.00
Remove Bold	0.00	0.00	0.00
Undelete Character	0.00	0.00	0.00
Totals	100.00	100.00	0.82

Table 3: WPS Keystroke Frequency Table

FROM TO-->	A/N	Left	Right	Delete	Down	Up	Return	Tab	Word	Ins/Ovr
A/N	.92	.00	.00	.03	.01	.00	.02	.00	.00	.00
Left	.03	.93	.01	.00	.01	.00	.00	.00	.00	.00
Right	.03	.01	.92	.01	.00	.00	.00	.00	.00	.01
Delete	.25	.00	.01	.70	.01	.00	.01	.01	.00	.00
Down	.02	.08	.03	.01	.78	.03	.01	.00	.00	.00
Up	.03	.04	.06	.00	.04	.78	.01	.00	.00	.00
Return	.38	.00	.02	.02	.04	.07	.19	.24	.00	.00
Tab	.52	.01	.01	.05	.03	.01	.00	.35	.00	.00
Word	.00	.02	.01	.00	.00	.00	.00	.00	.86	.00
Ins/Ovr	.42	.02	.06	.21	.06	.01	.02	.01	.00	.16

Table 4: EPT Keystroke Transition Probabilities ($n = 1$)

these statistics, we assigned the time preceding each keystroke to the keystroke in question. Analogous computations based on the time following each keystroke give very similar results.

Table 2 shows that typing accounted for about one half of editor usage; both in terms of keystroke frequency and time. The keystroke-level editing functions showed a non-uniform distribution of usage frequencies; the most frequently used functions are over 1000 times more frequent than the least frequently used functions. The top five editing functions account for more editing time than all the other functions combined. Four of these functions are the up, down, left, and right cursor movement functions — collective use of these functions accounted for about one quarter of user editing time. Single character deletion was the next most frequently used function, accounting for about 7% of users' time.

Table 3 shows summary statistics for the WPS work sample. The format of Table 3 is the same as that for Table 2. The table treats "gold" key editing functions as a single keystroke. Typing accounted for about one half of user time, as was the case with the EPT work sample. The distribution of times across functions is again highly non-uniform. The three most often used functions (Advance, Line, Backup) are all cursor movement functions, similar to the arrow function in EPT. These three functions account for just over 25% of total editing time. Single character deletion (Rub Character) was the next most frequent function.

3.3. Keystroke Transition Analysis

In addition to computing results at an individual keystroke level, we also examined transitions between keystrokes. Table 4 shows results of such analysis in the form of a keystroke transition matrix. The table shows keystroke transitions for the 10 most frequently used EPT functions. The function preceding the transition is shown along the left-hand margin. The function following the transition is shown along the top of the table. The entries in the matrix show the observed probability of transition from the row keystroke to the column keystroke. For example, the upper-left hand cell pertains to transitions between alphanumeric keys (printing characters). The transition probability of .92 indicates that if the user has just entered a printing character, the probability is .92 that the next keystroke will also be a printing character. The sum of all the probabilities in a row is always 1.0 (accounting for all transitions). Note that this is not true of the entries in Table 4; only the upper-left hand corner of the full 29 by 29 entry matrix is presented.

Table 5 shows a transition matrix similar to Table 4, except the upper entry in each cell shows the actual number of transitions of each type that took place. In addition to computing actual transitions, we also estimated how many transitions ought to occur, based on the assumption that the entries in the matrix are determined only by the marginal totals. The empirical matrices differ substantially from expectation, given such a hypothesis.

FROM TO--> / V	A/N	Left	Right	Delete	Down	Up	Return	Tab	Word	Ins/Ovr	Totals
	252543	745	1372	9072	2199	708	4487	1153	1	307	273504
A/N	146536	35279	34231	20100	15506	8628	3778	2774	922	915	
	2156	61052	648	298	485	310	150	105	3	304	65846
Left	35278	8494	8241	4839	3733	2077	909	668	222	220	
	1971	679	58923	774	350	254	190	32	10	402	63893
Right	34232	8242	7997	4696	3622	2016	882	648	215	214	
	9280	155	385	26173	536	135	288	214	1	153	37516
Delete	20100	4839	4695	2757	2127	1184	518	381	126	125	
	464	2347	965	311	22722	902	191	33	10	96	28936
Down	15503	3732	3622	2127	1641	913	400	293	98	97	
	420	659	1003	80	703	12637	104	59	10	52	16105
Up	8629	2077	2016	1184	913	508	222	163	54	54	
	2664	4	112	120	287	478	1371	1688	0	21	7045
Return	3775	909	882	518	399	222	97	71	24	24	
	2702	41	55	253	174	41	22	1830	0	8	5177
Tab	2774	668	648	380	294	163	72	53	17	17	
	4	28	23	5	2	6	3	0	1482	1	1720
Word	922	222	215	126	98	54	24	17	6	6	
	711	32	102	358	104	17	29	12	0	279	1707
Ins/Ovr	915	220	214	125	97	54	24	17	6	6	
Totals	273513	65850	63894	37518	28943	16105	7051	5178	1721	1707	510503

Key to entries:
observed number of transitions
expected number of transitions

Table 5: EPT Keystroke Transition Matrix, $n = 1$ (Totals from full matrix)

We were interested in describing the degree of organization, or degree of non-randomness, inherent in the users' keystroke patterns. Suppose, as an example, that keystroke sequences are entirely random. We would then expect the empirical transition frequencies to be very close to those that we would predict on the basis of the marginal frequencies. Clearly, this is not the case for the empirical data shown in Table 5. A comparison of obtained with expected frequencies shows the extent to which the actual keystroke behavior deviates from random expectation. We can compute a chi-square statistic using the expected and obtained frequencies. This shows the two matrices to be significantly different at the .001 level (chi square = 2,249,886; df = 273). Another way of expressing the same idea is with an index of predictive association [7, p. 747]. This is an index varying from 0.0 to 1.0, which shows the extent to which knowing the initial state improves the accuracy of prediction of a subsequent state. The index of predictive association for the data in Table 5 is 0.73 (with a 95% confidence limit of less than .01). For a transition matrix based on random key sequences, the index would approach 0.0.

We were also interested in attempting to describe the degree to which keystroke behavior is organized over longer sequences. Table 5 shows transitions between keystroke n and keystroke $n + 1$, for all n in our sample. We also computed transition matrices for transitions between keystrokes n and $n + m$ for m between 1 and 25. Table 6 shows such a matrix, in this case for $m = 12$. That is, the column keystrokes are those that follow the row keystrokes with a delay of 11 intervening strokes. The intent of this analysis is to see how far ahead in a sequence of strokes we can make good predictions, based on knowledge of the stroke at the start of the sequence.

We have generated expected numbers of keystrokes for the n to $n + 12$ transitions; these are also shown in Table 6. Notice that the actual frequencies still deviate substantially from the random predictions, but that the deviations are smaller than those in Table 5 (n to $n + 1$) predictions. This

Key to entries:
observed number of transitions
expected number of transitions

FROM / TO	A/N	Left	Right	Delete	Down	Up	Return	Tab	Word	Ins/Ovr	Totals
A/N	211716 146238	11597 35322	7323 33721	21920 20093	4312 13774	3086 8412	4599 3668	3394 2771	87 860	306 868	270025
Left	11385 35179	37604 8497	6316 8112	1885 4834	3518 3314	1947 2024	357 882	368 667	171 207	380 209	64958
Right	10686 33961	6115 8203	36242 7831	2864 4666	3118 3199	1568 1954	404 852	221 644	106 200	374 202	62708
Delete	22174 19994	1520 4829	2090 4610	8354 2747	972 1883	508 1150	452 502	425 379	19 118	79 119	36918
Down	3423 14872	4920 3592	5133 3429	684 2043	8500 1401	3198 855	257 373	125 282	109 87	129 88	27460
Up	1980 8479	2470 2048	3123 1955	318 1165	2960 799	3866 488	125 213	77 161	110 50	94 50	15657
Return	4613 3651	112 882	479 842	433 502	188 344	344 210	379 92	97 69	14 21	12 22	6742
Tab	3506 2763	264 667	172 637	487 380	64 260	81 159	96 69	370 52	1 16	13 16	5102
Word	74 887	139 214	104 205	22 122	149 84	96 51	10 22	4 17	702 5	7 5	1638
Ins/Ovr	455 878	239 212	339 203	103 121	176 83	83 51	9 22	12 17	2 5	158 5	1622
Totals	271408	65556	62584	37292	25564	15612	6808	5143	1596	1611	501150

Table 6: EPT Keystroke Transition Matrix, $n = 12$ (Totals from full matrix)

result is also expressed in the value of the index of predictive association which is 0.30 for the data in Table 6.

Figure 1 plots the index of predictive association as a function of the number of transitions between keystroke n and $n + m$ for $m = 1$ to 25. For EPT this function is plotted by circles. Notice the smooth decay of the the value of the function, indicating that it becomes progressively harder to predict the longer range one's prediction is.

The transition analysis described above was also performed for the WPS work sample. Table 7 shows the transition probabilities for the 10 most frequently used functions. The overall picture of editing that emerges from these data is very similar to that for EPT; the most frequent transitions are generally along the diagonal. On Figure 1 the predictive association for the WPS sample is depicted as triangles. The general shape of the curve is very close to that for the EPT work sample; it starts at a slightly lower level, crosses over, and appears to level off at a somewhat higher level, but the overall shape of the curve is almost identical.

3.4. State Analysis

In addition to analyzing our data at the level of individual keystrokes, we can define sequences of keystrokes as editing states and analyze the data at that level. This gives a more global picture of user behavior. A state consists of a sequence of keystrokes relating to only one editing function. For example, Type state is defined as an uninterrupted string of text. Cursor state is defined as any uninterrupted string of cursor positioning operations. Erase state involves any string of keystrokes concerned with deleting text. Edit state is eclectic but includes any continuous string of keystrokes concerned with altering or moving text. Start state is automatically entered when the editor is turned on; Finish is always the last state in an editing session. Pause state is entered after 150 seconds of user inactivity and ends with the next keystroke. Finally, Help state involves asking for on-line documentation. By definition no state may follow itself and no state may be nested within another state.

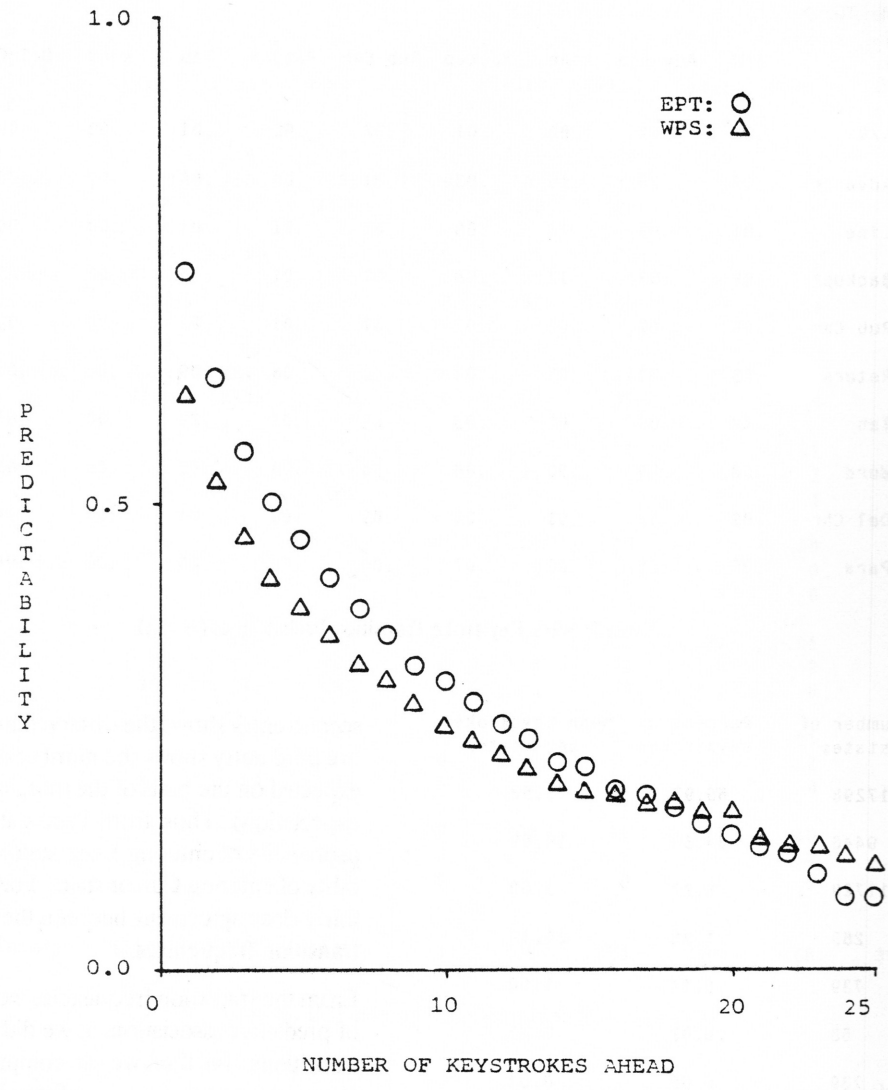

Figure 1: Predictability of Future Keystrokes as a Function of Lookahead Length

Table 8 shows the state analysis for EPT, including the number of occurrences of each state, the percent of keystrokes in each state and the mean number of keystrokes per state. The 739 Start states mean that there were 739 editing sessions in the work sample. On average, users in Type state tended to generate strings of 16.5 characters (this average is somewhat misleading, the frequency distribution is highly skewed with a mode of 1 character and a very long tail — the longest recorded string was 512 characters). Erase state consisted overwhelmingly of single-character-delete strokes so the 3.6 keystroke average is very close to the mean number of characters deleted per state.

We are not able to present an identical analysis for WPS as of this writing because it has a much more complex command syntax. However, the mean length of uninterrupted character strings was 17.2, very close to the 16.5 figure for the EPT sample. Table 9 shows the percentage of keystrokes for the WPS sample falling into some of the states defined for EPT. Even this incomplete comparison reveals a striking similarity of usage patterns.

3.5. Transitions Between States

We earlier discussed transitions between keystrokes. A similar analysis is possible at the state level. Table 10 shows a transition matrix for the EPT states. This matrix shows transitions between state n and state $n + 1$ for all states in the work sample. Notice that all the diagonal entries are zero; this follows from the definitions of the states themselves. For each cell in the matrix there are three entries. The first shows the row transition probability. The

FROM TO--> | V	A/N	Advance	Line	Backup	Rub Chr	Return	Tab	Word	Del Chr	Para
A/N	.92	.02	.00	.01	.02	.02	.01	.00	.00	.00
Advance	.03	.78	.10	.03	.01	.00	.00	.02	.00	.00
Line	.01	.09	.76	.08	.01	.01	.01	.00	.00	.00
Backup	.06	.03	.12	.68	.02	.01	.01	.03	.01	.00
Rub Chr	.30	.07	.00	.02	.57	.01	.01	.00	.00	.00
Return	.29	.08	.01	.03	.01	.38	.09	.00	.00	.00
Tab	.49	.09	.01	.03	.05	.01	.29	.00	.01	.00
Word	.08	.08	.00	.06	.08	.00	.02	.58	.01	.00
Del Chr	.07	.12	.01	.03	.01	.00	.01	.00	.74	.00
Para	.00	.08	.03	.07	.00	.00	.00	.00	.00	.73

Table 7: WPS Keystroke Transition Probabilities ($n = 1$)

State	Number of states	Percent of keystrokes	Mean keystrokes per state
Type	17298	55.97	16.53
Cursor	9446	34.39	18.60
Erase	11729	8.27	3.60
Edit	253	1.20	24.12
Finish	739	0.14	1.00
Help	56	0.02	2.27
Start	739	0.00	0.01
Pause	204	0.00	0.00

Table 8: EPT State Summary Table

State	Percent of total keystrokes
Type	59.16
Cursor	34.45
Erase	4.44
Other (includes Edit)	1.95

Table 9: WPS State Summary Table

second entry shows the observed number of transitions and the third entry shows the number of transitions to be expected on the basis of the marginal totals (random expectation). Thus, from Type state there is a .57 probability of entering Erase state next and a .41 probability of entering Cursor state. For the most part, there is fairly close agreement between the obtained and expected transition frequencies.

From the transition frequencies we can compute the index of predictive association, as we did for the keystroke transitions. Further, we can compute this index for transitions n to $n + m$, as before. Figure 2 shows the index for up to 25 states ahead. The index falls off slowly as a function of number of states ahead and the values appear to form an alternating series with even-numbered lookaheads consistently higher than adjacent odd-numbered lookaheads. Table 11 helps explain this pattern. It shows a transition matrix for transitions between states n and $n + 2$. Notice the high transition probabilities along the diagonal and the corresponding high discrepancy between the observed and expected values along the diagonal. This same pattern is evident in the other even-numbered transition matrices. It means that when users leave a state, they tend to return directly to it after they have entered a single intervening state.

FROM TO--> / V	Type	Erase	Cursor	Start	Finish	Edit	Pause	Help	Totals
	0	.57	.41	.00	.02	.00	.00	.00	
Type	0	9800	7032	0	336	13	89	22	17292
	0	9355	7129	0	476	163	132	36	
	.86	.00	.13	.00	.01	.00	.00	.00	
Erase	10056	0	1551	0	87	9	20	6	11729
	9355	0	2133	0	142	49	39	11	
	.74	.20	.00	.00	.02	.02	.01	.00	
Cursor	7024	1847	0	0	239	224	95	17	9446
	7130	2133	0	0	108	37	30	8	
	.15	.06	.73	.00	.04	.01	.00	.01	
Start	112	46	537	0	31	6	0	7	739
	476	142	108	0	7	2	2	1	
	.00	.00	.00	.00	.00	.00	.00	.00	
Finish	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	
	.11	.06	.80	.00	.02	.00	.00	.01	
Edit	28	15	203	0	5	0	0	2	253
	163	49	37	0	2	0	1	0	
	.31	.09	.41	.00	.19	.00	.00	.01	
Pause	63	18	83	0	38	0	0	2	204
	132	39	30	0	2	1	0	0	
	.16	.05	.71	.00	.05	.02	.00	.00	
Help	9	3	40	0	3	1	0	0	56
	36	11	8	0	1	0	0	0	
Totals	17292	11729	9446	0	739	253	204	56	39719

Key to entries:
transition probabilities
observed number of transitions
expected number of transitions

Table 10: EPT Editing State Transition Matrix ($n = 1$)

4. Discussion

4.1. General Conclusions

A number of general conclusions about screen-oriented text editing are suggested by the results. The distribution of use across editing functions is grossly non-uniform. Certain functions are used constantly, others almost never. A rough rule of thumb that emerges from our data is that free usage text editing consists of about 1/2 typing, about 1/4 cursor movement, about 1/8 deletion, and 1/8 all other functions put together. This rule has been shown to apply in two different situations: knowledge workers creating documents with an experimental text editor, and secretaries transcribing and updating documents on a commercial word processing system. Considering the differences in the editors, the workers, and the tasks, the correspondence between the work samples is striking. It implies that it may be possible to optimize a single editor for a variety of users.

Another finding to emerge from the state analysis is that users tend to alternate editing subtasks; they tend to return to the state they left one state before. For example, if a user is typing, he tends to leave that state to do one other thing (perhaps deleting) and then immediately returns to typing.

4.2. Extending the State Analysis

The set of states presented here was sufficient for analyzing EPT, but not for analyzing more powerful editors. Two new states, Format and Manage, can handle these more sophisticated functions. Format includes formatting commands that alter the outward appearance of a document. Manage includes commands for managing multiple files and windows while within an editor.

Since EPT was a simple editor, it was easy to build the necessary state analysis programs. State analysis is much more difficult than keystroke analysis for any large production editor due to the extensive parsing that must be done. This problem is alleviated in situations where the analysis can be done using the code from the editor's parser.

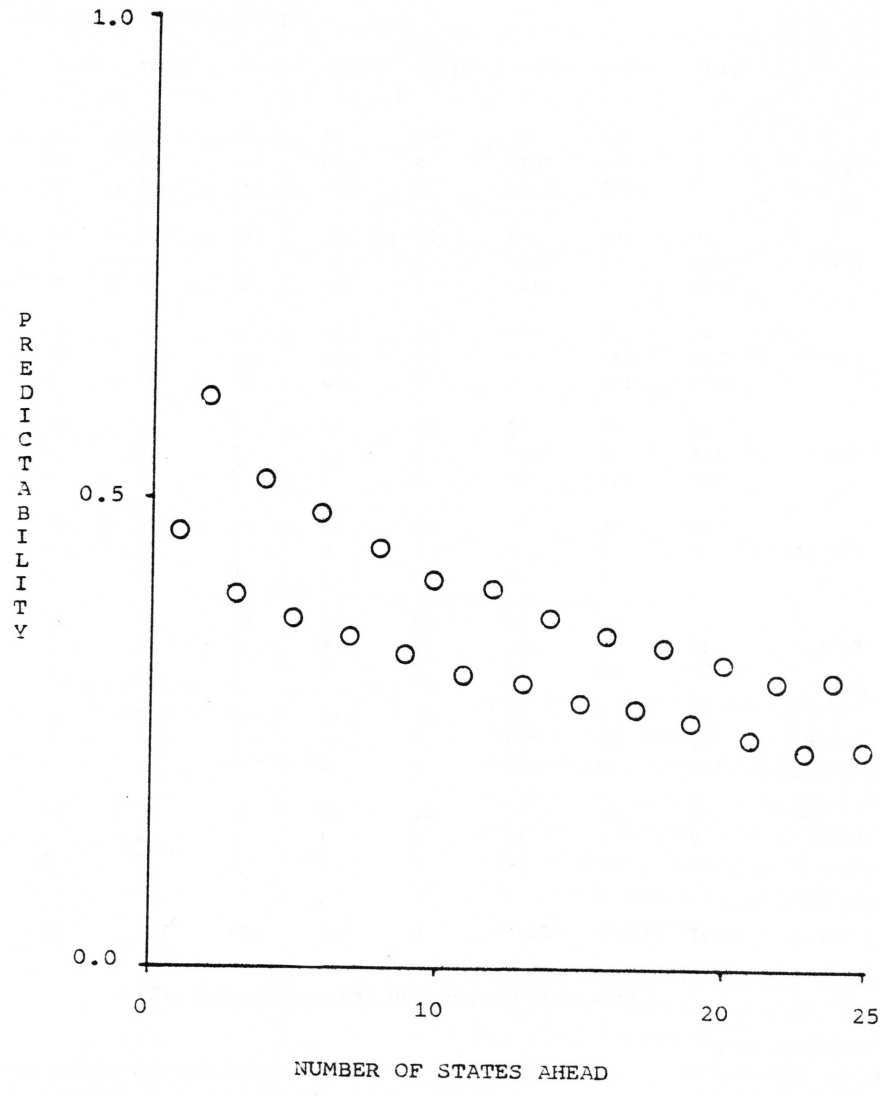

Figure 2: Predictability of Future States as a Function of Lookahead Length

4.3. Relation to Past Research

The analysis gives us a picture of editing based on actual usage data, rather than on some abstract model. As such, the data presented here provide a good test of the relevance of these abstract models and can be seen as a complement to them.

4.4. Implications for Editor Design

Data such as these can be used to drive the software architecture of text editors. Raw frequency-of-use data indicates which aspects of editor design should be optimized. For instance, cursor placement is the single most time consuming editing function next to typing itself. Thus, the mechanics of cursor positioning deserve considerable engineering attention and optimization. In addition, the transition data might be used to tune editor software to the behavior patterns of users. For example, faster performance could be obtained by reordering the alternatives within a parser, and by improving the locality of reference within the editor.

4.5. Implications for Keyboard Design

Beyond its implications for editor design, this data provides a sound empirical basis for the application of McCormick's [8] principles of keyboard design. In order to place keys in terms of either their overall usage frequency or the frequency of transitions between keystrokes one must have data of the type provided here.

FROM TO-->	Type	Erase	Cursor	Start	Finish	Edit	Pause	Help	Totals
	.88	.07	.03	.00	.01	.01	.00	.00	
Type	14936	1117	514	0	214	100	61	14	16956
	7473	5082	3875	0	308	107	89	21	
	.07	.73	.17	.00	.02	.00	.00	.00	
Erase	837	8539	1934	0	219	38	62	13	11642
	5131	3489	2661	0	211	74	61	15	
	.09	.19	.69	.00	.02	.00	.00	.00	
Cursor	788	790	6329	0	229	11	42	18	9207
	4058	2760	2104	0	167	58	48	12	
	.62	.19	.10	.00	.04	.02	.03	.00	
Start	440	137	68	0	28	13	21	1	708
	312	212	162	0	13	4	4	1	
	.00	.00	.00	.00	.00	.00	.00	.00	
Finish	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	
	.37	.14	.13	.00	.02	.32	.01	.01	
Edit	91	36	32	0	4	80	3	2	248
	109	74	57	0	5	2	1	0	
	.40	.28	.16	.00	.08	.00	.08	.00	
Pause	67	46	26	0	14	0	13	0	166
	73	50	38	0	3	1	1	0	
	.40	.34	.11	.00	.00	.09	.04	.02	
Help	21	18	6	0	0	5	2	1	53
	23	16	12	0	1	0	0	0	
Totals	17180	11683	8909	0	708	247	204	49	38980

Key to entries:
transition probabilities
observed number of transitions
expected number of transitions

Table 11: EPT Editing State Transition Matrix ($n = 2$)

4.6. Implications for Benchmark Tests

Benchmarking is an appropriate testing method for text editors when there is an interest in making comparisons among existing editors, or improving the performance of an editor which may be under development. The best way one can develop a realistic set of benchmark tasks for a text editor is on the basis of actual usage statistics. Roberts has made the first attempt at developing such benchmark tasks. These tasks include a test of speed of use for experts, developed on the basis of an informal study.

In order to evaluate the representativeness of Roberts' benchmarks two EPT experts performed Roberts' expert tasks. Their data was then compared to real EPT behavior in Table 12. Table 12 shows that users spent much more time in the Edit and Cursor states when performing Roberts' tasks than they did in free use.

State	Actual Usage	Roberts' Task
Type	53%	14%
Cursor	32%	49%
Erase	10%	11%
Finish	3%	1%
Edit	2%	25%
Help	0%	0%
Start	0%	0%

Table 12: EPT State Usage as Percentage of Working Time

An increased knowledge of actual user behavior should enable researchers to develop benchmarks which more closely reflect actual behavior. In addition, given the similarities in user behavior, newly developed benchmarks should be widely applicable.

References

1. Card, Stuart K., Moran, Thomas P., and Newell, Allen. Computer text-editing: an information processing analysis of a routine cognitive skill. *Cognitive Psychology 12* (1980), 32–74.

2. Card, Stuart K., Moran, Thomas P., and Newell, Allen. The keystroke-level model for user performance with interactive systems. *Comm. ACM 23* (1980), 396–410.

3. Embley, David W. and Nagy, George. Empirical and formal models for the study of computer editors. In *Computing Skills and the User Interface*, M. Coombs and J. L. Alty, Eds., Academic Press, London, 1981.

4. Embley, David W. and Nagy, George. Behavioral aspects of text editors. *Comput. Surv. 13* (1981), 33–70.

5. Hammer, John M. The human as a constrained optimal text editor. Ph.D. Th., University of Illinois, June, 1981.

6. Hammer, John M. and Rouse, W. B. Analysis and modeling of freeform text editing behavior. Proc. 1979 Internat. Conf. Cybernetics and Society, Oct., 1979, pp. 659–663.

7. Hays, William L. *Statistics for the Social Sciences*. Holt Rinehart Winston. New York, 1973. Second Edition.

8. McCormick, Ernest J. *Human Factors in Engineering and Design*. McGraw-Hill, New York, 1976. Fourth Edition.

9. Roberts, Teresa L. Evaluation of computer text editors. Report SSL-79-9, Xerox PARC, Nov., 1979. Stanford Ph.D. dissertation.